(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 March 2020 (19.03.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/053469 A1

(51) International Patent Classification:
G06F 8/65 (2018.01) *G06F 21/57* (2013.01)

(21) International Application Number:
PCT/FI2018/050647

(22) International Filing Date:
12 September 2018 (12.09.2018)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **NOKIA SOLUTIONS AND NETWORKS OY** [FI/FI]; Karakaari 7, 02610 Espoo (FI).

(72) Inventors: **OLIVER, Ian Justin**; Hitåvägen 153, 01150 Söderkulla (FI). **LIMONTA, Gabriela**; Rantaharju 10 B 42, 02230 Espoo (FI). **VIGMOSTAD, Borger Ormiskangas**; Itämerenkatu 18 A 3, 00180 Helsinki (FI). **KALLIOLA, Aapo**; Pyynikintie 4-8 A 10, 00710 Helsinki (FI).

(74) Agent: **NOKIA TECHNOLOGIES OY** et al.; Ari Aarnio, IPR Department, Karakaari 7, 02610 Espoo (FI).

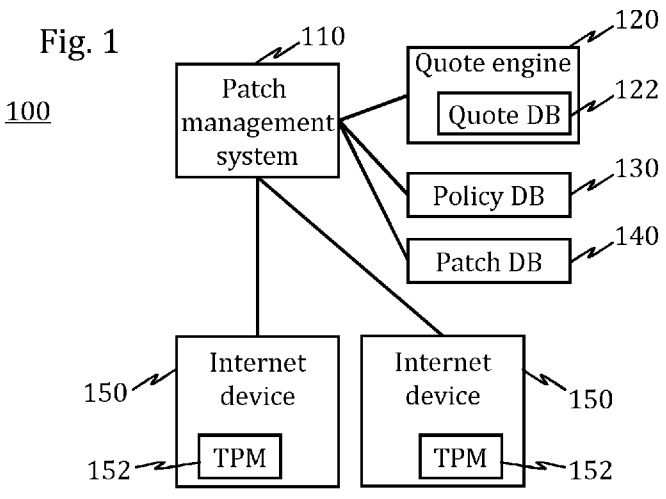
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report (Art. 21(3))*

(54) **Title:** METHOD AND APPARATUS FOR AUTOMATING SOFTWARE UPDATING OF DEVICES

Fig. 1



(57) **Abstract:** An update system has a communication block to communicate with the Internet devices; computer program code; and a processing block to execute the computer program code and accordingly to perform: checking states of a plurality of Internet devices through quotes obtained from trusted platform modules of the Internet devices; checking patch availability for each of the plurality of Internet devices; making patching decisions independent of other entities such as data processing agents; determining the need for software updates based on quotes; and maintaining policies that drive the quoting process.

WO 2020/053469 A1

METHOD AND APPARATUS FOR AUTOMATING SOFTWARE UPDATING OF DEVICES

TECHNICAL FIELD

[0001] Various example embodiments relate to automating software updating of devices.

BACKGROUND

[0002] This section illustrates useful background information without admission of any technique described herein representative of the state of the art.

[0003] Networked devices, such as computer servers, need periodic updates. These updates, from now on referred to as patches, should be applied to all devices in a timely manner. Patching of systems is often done in a manual and time-consuming manner.

[0004] Various processor operated devices are typically shipped with a set of firmware and software. After some time, the devices need to be updated by means of software patches. These updates may occur regularly. They can include firmware, software and application updates. Regular updates keep the devices protected against vulnerabilities in software.

[0005] There are devices that are easily left without updates. For example, there is an increasing number of Internet of Things (IoT) devices that can be connected to the cloud and Internet enabled toys and entertainment devices that their users cannot or do not remember to update. The Internet of Things devices are devices that automatically communicate measurements or detections of physical events to other devices or that control physical actuators according to commands received from the Internet, without need for manual supervision or interaction for such tasks.

[0006] The updates of Internet enabled devices are not only needed to protect the devices in question from abuse, but for avoiding their use in botnet attacks against other Internet devices. Such attacks may harm other devices and at the very least, they waste bandwidth and may trigger blacklisting the devices and closing Internet access by Internet service providers.

[0007] Personal computers typically periodically poll for available updates to the operating system and install the updates automatically or with a user permission.

This operation mode is rather well scalable as the detection of patching need is distributed to the computers themselves. However, large number of computers skip updates for various reasons including the users' choice not to approve any updates, e.g., in fear of some application or service stopping to work due to buggy patches or changing way of operation after feature updates.

SUMMARY

[0008] Various aspects of examples of the invention are set out in the claims.

[0009] According to a first example aspect of the present invention, there is provided an update system comprising:

[0010] a communication block configured to communicate with Internet devices comprising trusted platform modules;

[0011] computer program code; and

[0012] a processing block configured to execute the computer program code and accordingly to perform:

[0013] checking states of a plurality of Internet devices through quotes obtained from trusted platform modules of the Internet devices;

[0014] checking patch availability for each of the plurality of Internet devices;

[0015] making patching decisions independent of other entities;

[0016] determining the need for software updates based on quotes; and

[0017] maintaining policies that drive the quoting process.

[0018] The entities may be or comprise data processing agents in communicative connection with the update system. The entities may be or comprise data processing agents in the update system.

[0019] The determining of the need for software updates based on the quotes may be made by the patch management system.

[0020] Each of the Internet devices may comprise a trusted platform module. Alternatively, some of the Internet devices may comprise a trusted platform module.

[0021] The policies may allow an agent, e.g. a patch management system, to determine which Internet device needs an update.

[0022] The checking of the states of the plurality of Internet device may comprise checking for each of the Internet devices an update state for each of one or more updateable components. The checking of the update state may comprise checking

of current software or firmware version.

[0023] The communication block may comprise computer program code. The communication block may comprise a communication circuitry.

[0024] The processing block may comprise computer program code. The processing block may comprise a processing circuitry.

[0025] The received quotes may comprise platform configuration registers. The

[0026] The processing block may be further configured to define the policies that drive the quoting process.

[0027] The update system may comprise or the update system may be a patch management system. The patch management system may comprise the communication block; the processing block; and at least some of the computer program code. The update system may further comprise a policy database comprising the policies that drive the quoting process. The update system may further comprise a quote engine configured to provide the patch management system with the predicted quotes. The quote engine may comprise a quote database comprising previously computed predicted quotes for different Internet device components and patches. The quote engine may further comprise a quote processing block configured to compute predicted quotes on request of the patch management system. The quote engine may further be configured to update the quote database on computing new predicted quotes.

[0028] The updating system may further comprise a patch database. The patch database may comprise patches for the components of the Internet devices. Alternatively, or additionally, the patch database may comprise links to externally stored patches for the components of the Internet devices.

[0029] The software element may comprise a basic input/output system program code. The software element may comprise firmware of a hardware component of the Internet device. The software element may comprise a device driver. The software element may comprise a software library component, such as a dynamic link library component. The software element may comprise at least one operating system component. The software element may comprise at least one application component.

[0030] The processing block may be further configured to perform verifying

each of the Internet devices has been successfully patched. The verifying may comprise comparing a received quote of the Internet device with a predicted quote. The predicted quote may comprise check a code produced for a software element being patched after a patch in question has been applied to the software element. The check code may be based on a cyclic redundant code. The check code may comprise an authenticator. The authenticator may comprise a cryptographic presentation of the check code. The cryptographic presentation may comprise a cryptographic one-way hashing of the check code. The authenticator may comprise a digital signature of the check code. The digital signature may be formed using an attestation key of the Internet device. The attestation key may be formed by a trusted platform module of the Internet device. The authenticator may comprise as such or as a derivative state information of the trusted platform module, such as a reboot count. The authenticator may comprise as such or as a derivative a timestamp.

[0031] The verifying of successful patching may comprise obtaining a new quote from an Internet device and determining whether the new quote matches a predicted quote. If no, a patching error may be indicated. If yes, the verifying may further comprise determining whether reboot count has increased by an expected amount. If no, a patching error may be indicated. If yes, the verifying may further comprise determining whether the timestamp of the new quote indicates lapse of time by an expected duration. If no, a patching error may be indicated. If yes, the patching success may be considered correctly applied.

[0032] Some of the patches may be made unnecessary by subsequent patches so that updating of one Internet device with one patch may bring the Internet device up to date. The verifying may then indicate successful patching regardless whether all preceding patches have been applied by the Internet device.

[0033] The processing block may be further configured to maintain statistics of average success rate of different patches. The statistics may indicate the average success rate separately for different classes of Internet devices. The different classes may be classified using at least one of: type; make; model; model version; and age. The processing block may be further configured to alert of patches that statistically appear failure prone. The alerting may inform of the class or classes in which the patching statistically appears failure prone. The failure prone patch may have a success rate less than any one of: 99%; 95 %; 90 %; 80 %; 70%; 60 % or 50 %.

[0034] The update system may comprise or the update system may be an attestation server. The patch management system may be configured to communicate with the attestation server. The patch management system may be configured to communicate with the Internet devices directly or via the attestation server.

[0035] The update system may comprise a blockchain delivery channel configured to distribute patches to the Internet devices through a blockchain. The Internet devices may obtain the patches from the blockchain.

[0036] Any one of the policy database, quote database; and patch database may be centralized or distributed. Centralized databases may be implemented using a relational database, such as a structured query language database. The distributed databases may be implemented using one or more distributed relational databases and/or a blockchain, such as Ethereum.

[0037] The update system may comprise a first subset of the plurality of Internet devices.

[0038] According to a second example aspect of the present invention, there is provided an update system comprising:

[0039] a communication block configured to communicate with Internet devices comprising trusted platform modules;

[0040] computer program code; and

[0041] a processing block configured to execute the computer program code and accordingly to perform:

[0042] checking states of a plurality of Internet devices through quotes obtained from trusted platform modules of the Internet devices;

[0043] checking patch availability for each of the plurality of Internet devices;

[0044] making patching decisions independent of other agents in the system;

[0045] determining the need for software updates based on quotes; and

[0046] maintaining policies that drive the quoting process.

[0047] According to a third example aspect of the present invention, there is provided a method comprising:

[0048] communicating with Internet devices comprising trusted platform modules;

[0049] checking states of a plurality of Internet devices through quotes obtained from trusted platform modules of the Internet devices;

[0050] checking patch availability for each of the plurality of Internet devices;

[0051] making patching decisions independent of other entities;

[0052] determining the need for software updates based on quotes; and

[0053] maintaining policies that drive the quoting process.

[0054] The determining of the need for software updates based on the quotes may be made by the patch management system.

[0055] According to a fourth example aspect of the present invention, there is provided a computer program comprising computer executable program code configured to execute any method of the third example aspect.

[0056] The computer program may be stored in a computer readable memory medium. The computer medium may be a non-transitory memory medium.

[0057] Any foregoing memory medium may comprise a digital data storage such as a data disc or diskette, optical storage, magnetic storage, holographic storage, opto-magnetic storage, phase-change memory, resistive random access memory, magnetic random access memory, solid-electrolyte memory, ferroelectric random access memory, organic memory or polymer memory. The memory medium may be formed into a device without other substantial functions than storing memory or it may be formed as part of a device with other functions, including but not limited to a memory of a computer, a chip set, and a sub assembly of an electronic device.

[0058] Different non-binding example aspects and embodiments of the present invention have been illustrated in the foregoing. The embodiments in the foregoing are used merely to explain selected aspects or steps that may be utilized in implementations of the present invention. Some embodiments may be presented only with reference to certain example aspects of the invention. It should be appreciated that corresponding embodiments may apply to other example aspects as well.

BRIEF DESCRIPTION OF THE DRAWINGS

[0059] For a more complete understanding of example embodiments of the present invention, reference is now made to the following descriptions taken in connection with the accompanying drawings in which:

[0060] Fig. 1 shows an architectural drawing of a system of an example embodiment;

[0061] Fig. 2 shows a block diagram of the patch management system of Fig.

1;

[0062] Fig. 3 shows a block diagram of the Internet device of Fig. 1;

[0063] Fig. 4 shows a flow chart of a process of an example embodiment;

[0064] Fig. 5 shows a flow chart of a process of an example embodiment;

[0065] Fig. 6 shows a signaling chart of an example embodiment;

[0066] Figs. 7 and 8 show schematic architectural drawings of two different example embodiments; and

[0067] Fig. 9 shows a flow chart of a process of an example embodiment.

DETAILED DESCRIPTON OF THE DRAWINGS

[0068] An example embodiment of the present invention and its potential advantages are understood by referring to Figs. 1 through 9 of the drawings. In this document, like reference signs denote like parts or steps.

[0069] Fig. 1 shows an architectural drawing of an update system 100 of an example embodiment The update system 100 comprises a patch management system 110 and connected to the patch management system: a quote engine 120 comprising a quote database 122; a policy database 130; a patch database 140; and a plurality of Internet devices 150 which comprise trusted platform modules 152. Any one or more of the quote engine 120; quote database 122; policy database 130; and patch database 140 may be entirely or partly comprised by the patch management system 110.

[0070] Fig. 2 shows a block diagram of the patch management system of Fig. 1. The patch management system 110 comprises:

[0071] a communication block 210 configured to communicate with the Internet devices 150;

[0072] computer program code 222 stored in a memory 220; and

[0073] a processing block 230 configured to execute the computer program code and accordingly to perform:

[0074] checking states 410 (Fig. 4) of a plurality of Internet devices through quotes obtained from trusted platform modules of the Internet devices;

[0075] checking patch availability 420 (Fig. 4) for each of the plurality of Internet devices;

[0076] making patching decisions 430 (Fig. 4) independent of other entities (e.g., data processing agents in the update system or in communicative connection with

the update system);

[0077] determining 440 (Fig. 4, e.g. by the patch management system) the need for software updates based on quotes; and

[0078] maintaining policies 450 (Fig. 4) that drive the quoting process.

[0079] In an example embodiment, each of the Internet devices 150 comprises a trusted platform module TPM. In an alternative embodiment, some of the Internet devices 150 comprise a TPM.

[0080] Fig. 3 shows a block diagram of the Internet device 150. The Internet device 150 comprises a communication block 310 configured to communicate over the Internet; a memory 310; one or more applications 322; one or more pieces of firmware 324; and an operating system 326. The memory 310 may comprise one or more non-volatile memories, including, for example, a hard disk drive or solid state drive of the Internet device 150; one or more non-volatile memories of various components such as of a graphics processing unit, or of a communication unit or of a microprocessor. The internet device 150 further comprises a basic input/output system 328 (which refers here to the BIOS code that is updateable rather than the hardware) and the trusted platform module 152. The trusted platform module 152 comprises a certificate 340; an endorsement key EK 350; an attestation key AK 360; and a plurality of platform configuration registers PCR 370. In an example embodiment, the PCRs 370 store measurements, e.g., in the form of hashes, of different software, firmware or BIOS components of the Internet device 150.

[0081] The attestation key AK 360 is in an example embodiment an attestation key complying with TPM 2.0. In another example embodiment, the attestation key AK 360 is in an example embodiment an attestation identity key complying with TPM 1.0. In an example embodiment, the certificate 340 is, or comprises, an attestation certificate compliant with TPM 2.0. In an example embodiment, the certificate 340 is, or comprises, an attestation identity key certificate compliant with TPM 1.0.

[0082] In an example embodiment, the TPM 152 stores cryptographic keys, certificates and confidential data. For example, the TPM 152 stores two unique key pairs: The EK 350 and the AK 360. Additionally, the platform configuration registers 370 store measurements, e.g., in the form of hashes, of the (updateable) components of the Internet device 150. In an example embodiment, the TPM 152 can be asked to provide a quote for a set of PCRs. The quote is, for example, a hash over the stored

values of the selected set PCRs. In an example embodiment, the TPM will then return the quote for the requested PCRs, a cryptographic signature of that quote (signed by the AK) and other information, including a timestamp and a reboot count.

[0083] In an example embodiment, the Internet device 150 signs the quote with the AK 360. In an example embodiment, the TPM 152 can also store an externally provided key, which has the AK of the Internet device 150 in a signature chain. This externally provided key can be used to sign quotes.

[0084] Fig. 4 has already been discussed in connection with Fig. 2 description.

[0085] In an example embodiment, the policies allow an agent, e.g. the patch management system 110, to determine which Internet device 150 needs an update, as will be further exemplified with reference to Figs. 5 and 6. In this document, the internet device 150 may refer to any device that uses the Internet for communication.

[0086] Fig. 5 shows a flow chart of a process of an example embodiment. The process comprises checking 510 whether patching is needed for the Internet devices 150. Let us consider this process for a given internet device 150. If not, it is determined that no patch is needed in step 580.

[0087] In an example embodiment, the checking of the states of the plurality of Internet device 150 comprises checking for each of the Internet devices 150 an update state for each of one or more updateable components. The checking of the update state may comprise checking of current software or firmware version.

[0088] In step 520 it is calculated which patch is needed. To this end, combinations of the present state of the Internet device 520 and of a set of patches P is generated 530 and an expected result V(E+P) is calculated 540 for the tested combination of present state and the set of patches P. It is then tested 550 if the tested combination would comply with current policy (e.g., correspond to the set of latest patches that are available for the Internet device 150). If a match is found, the tested set of patches P is presented 570 for patching the Internet device 150. Otherwise the process returns to step 530 until no more combinations of patch sets are available for the Internet device 150 and the process goes to end in step 560 (for the Internet device 150 in question). Notably, the process can be performed in plurality of parallel processes for individual Internet devices 150. Moreover, in an example embodiment, a group of Internet devices with same hardware and installed applications processed simultaneously as one group in terms of determining the need of patching and the

suitable patch or patches if any are needed. The actual patching and verifying the end result can be performed individually or on a group level as well, provided that that actual verification of successful patching can be made for each Internet device 150 of a group.

[0089] In an example embodiment, the policy database 130 comprises the policies that drive the quoting process. In an example embodiment, the updating system further comprises the quote engine 120 configured to provide the patch management system 110 with the predicted quotes. In an example embodiment, the quote engine 120 comprises the quote database 122 comprising previously computed predicted quotes 122 (or expected values, when the received quotes and the predicted quotes are in the form of values) for different Internet device 150 components and patches. In an example embodiment, the quote engine 120 further comprises a quote processing block (not shown) configured to compute predicted quotes on request of the patch management system 110. In an example embodiment, the quote engine 120 is further configured to update the quote database 122 on computing new predicted quotes.

[0090] In an example embodiment, the policy is or the policy comprises a set of expected values (or predicted quotes) for different measurements that can be taken from the Internet device 150. When the Internet device 150 is equipped with the TPM 152, the policy can be a mapping between the PCRs and expected results or values. For example, an administrator can define policies for an Internet device 150 to be considered in a trusted state. When attestation is done, the measurements or received quotes are checked against the expected values in the policy or expected quotes.

[0091] Furthermore, the policy can potentially become quite complex. Both simple and compound policies can be contemplated. A simple policy is a set of expected values. In an example embodiment, the policy is a compound policy that comprises set of simple policies. It is possible to define very specific policies as building blocks for more complex policies. For example: if there are two Internet devices 150 running the same BIOS version but different software on top, there is defined in an example embodiment a simple policy for the BIOS and then two different policies for the other software on top. Two compound policies can then be created, which compound policies will both share the BIOS policy. The quotes can also be requested separately for each separately updateable component so that the need of updating and the success can be

determined and the actual performing of updates in the Internet devices 150 can be verified.

[0092] In an example embodiment, the updating system 100 further comprises the patch database 140. In an example embodiment, the patch database 140 comprise patches for the components of the Internet devices 150. Alternatively, or additionally, the patch database 140 comprises links to externally stored patches for the components of the Internet devices 150.

[0093] In an example embodiment, the processing block 230 is configured to perform verifying that each of the Internet devices 150 has been successfully patched. In an example embodiment, the verifying comprises comparing a received quote of the Internet device 150 with the predicted quote. In an example embodiment, the predicted quote comprise check a code produced for a software element being patched after a patch in question has been applied to the software element. In an example embodiment, the check code is based on a cyclic redundant code. In an example embodiment, the check code comprises an authenticator. In an example embodiment, the authenticator comprises a cryptographic presentation of the check code such. In an example embodiment, the cryptographic presentation comprises a cryptographic one-way hashing of the check code. In an example embodiment, the authenticator comprises a digital signature of the check code. In an example embodiment, the digital signature is formed using an attestation key of the Internet device 150. In an example embodiment, the key is formed by the trusted platform module 152 of the Internet device 150. In an example embodiment, the authenticator comprises as such or as a derivative state information of the trusted platform module 152, such as a reboot count. In an example embodiment, the authenticator comprises as such or as a derivative a timestamp.

[0094] In an example embodiment, the verifying of successful patching comprises obtaining a new quote from an Internet device 150 and determining whether the new quote matches a predicted quote. If no, a patching error may be indicated. If yes, the verifying may further comprise determining whether reboot count has increased by an expected amount. If no, a patching error may be indicated. If yes, the verifying may further comprise determining whether the timestamp of the new quote indicates lapse of time by an expected duration. If no, a patching error may be indicated. If yes, the patching success may be considered correctly applied.

[0095] Some of the patches may be made unnecessary by subsequent patches so that updating of one Internet device 150 with one patch may bring the Internet device 150 up to date. The verifying may then indicate successful patching regardless whether all preceding patches have been applied by the Internet device 150.

[0096] In an example embodiment, the processing block 230 is further configured to maintain statistics of average success rate of different patches. The statistics may indicate the average success rate separately for different classes of Internet devices 150. In an example embodiment, the different classes are classified using at least one of: type; make; model; model version; and age. In an example embodiment, the processing block 230 is further configured to alert of patches that statistically appear failure prone. In an example embodiment, the alerting informs of the class or classes in which the patching statistically appears failure prone. The failure prone patch may have a success rate less than any one of: 99%; 95 %; 90 %; 80 %; 70%; 60 % or 50 %.

[0097] Fig. 6 shows a signaling chart of an example embodiment. The signaling chart shows the following events:

[0098] 610. The Internet device 150 requests an update policy from the policy DB 130.

[0099] 615. The policy DB returns the policy for PCRs 1 to 7 of the Internet device 150.

[0100] 620. The TPM 152 computes a quote against policy 620.

[0101] 625. The Internet device 150 uploads to the Quote DB 122 a quote for the PCRs 1-7.

[0102] 630. The Quote DB 122 is requested for the quote by the patch management system 110.

[0103] 635. The Quote DB 122 returns the requested quote to the patch management system 110.

[0104] 640. The patch management system 110 requests the policy from the policy DB 130.

[0105] 645. The policy DB 130 returns the requested policy.

[0106] 650. The patch management system 110 checks the PCRs 1-7 against the policy EV.

[0107] 655. The patch management system 110 requests a set of patches is

from the patch DB 140.

[0108] 660. The patch DB 140 returns the requested set of patches.

[0109] 665. The patch management system 110 tests whether the returned set of patches would comply with the policy EV. If yes, the set of patches is presented 680 by the patch management system 110 to the Internet device 150. If no, it is checked if there is another quote + set of patches to be tested in steps 655 to 670, and if there is, a new round is started in step 675, otherwise it is determined that the Internet device 150 is not to be patched with respect to its components corresponding to the PCRs 1-7 of this example.

[0110] In an example embodiment, the update system 100 comprises an attestation server 710 (Fig. 7). The patch management system 110 may be configured to communicate with the attestation server 710. The patch management system 110 may be configured to communicate with the Internet devices 150 directly (as in Fig. 7) or via the attestation server 710 (as in Fig. 8).

[0111] In an example embodiment, the update system 100 comprises a blockchain delivery channel configured to distribute patches to the Internet devices 150 through a blockchain. The Internet devices 150 may obtain the patches from the blockchain.

[0112] Any one of the policy database 130, quote database 122; and patch database 140 may be centralized or distributed. Centralized databases may be implemented using a relational database, such as a structured query language database. The distributed databases may be implemented using one or more distributed relational databases and/or a blockchain, such as Ethereum.

[0113] In an example embodiment, the update system 100 comprises a first subset of the plurality of Internet devices 150.

[0114] Fig. 8 shows a schematic architectural drawing of an example embodiments. Fig. 8 presents an attestation Server 810, which in this embodiment is a main component in which required functionality is encapsulated, i.e.: the mechanisms for requesting a quote, receiving a quote, cross-referencing with external systems (e.g., patch management system 850, a database 820 for storing relevant attestation data) and a reasoning or decision engine 818 that can reason in some form over the collected data and inputs.

[0115] The attestation server 810 of Fig. 8 comprises internal components

that include Quoting 812 that is a functionality for performing operations for the process of quoting a device.

[0116] The internal components further include an internal database 814, which is, for example, a functionality that performs operations for storing information related to attestation. This may additionally call external database 820 processing for convenience, long-term storage etc.

[0117] NB: in the diagram an external Database 820 is shown, which is, for example, be any external mechanism for storing data, including traditional databases, log files, blockchain, etc.

[0118] The internal components further include monitoring 816, which is, for example, a functionality that performs operations relating to active or passive monitoring of systems.

[0119] The internal components further include decision maker 818, which is, for example, a functionality that performs operations relating to deciding on the course of action of some event (internal or external) related to attestation functionality. Such functionality might be relatively simple and hard encoded in some code or could be implemented using some AI mechanism.

[0120] The internal components further include action 819, which is, for example, a functionality that performs or triggers some attestation action to take place. This could include communication with external systems.

[0121] Fig. 8 shows a box representing machines 840, which may include any devices that can receive (including proxy) a request for attestation, or can (including proxy) relay attestation data, e.g., in the form of a TPM2 quote or other similar structure.

[0122] Fig. 8 further shows a patch management system 850, which may be, e.g., any system that coordinates, manages, orchestrates etc. the update and patching of machines, software and other devices (IoT, Edge, etc.).

[0123] Fig. 8 further shows a security orchestrator 830. This may be, e.g., any system that coordinates, manages, orchestrates etc. security policies and response mechanisms to certain system events, e.g., attestation failures. In an example embodiment, the security orchestrator 830 may also interact with the attestation server 810 to obtain information about a given machine or processes' state as necessary. The security orchestrator may also be referred to as security orchestration

management.

[0124] Fig. 8 further shows a box for other systems 860, which may comprise any other system(s) that can receive or send pertinent information or signals to the attestation server 810. For example, in the proof of concept individual machines can report their boot status. This information can be used by the attestation server 810 to trigger a post-boot attestation of that machine. In another example, this is a virtual infrastructure manager (VIM) 870 in some network functions virtualization management and orchestration (NFV MANO) component that "proxies" or triggers the attestation server 810 on some operation in some way.

[0125] One Use Case is next presented:

[0126] The patch management system 850 notifies the VIM that a given machine 840 requires an update.

[0127] The VIM 870 requests that the attestation server 810 take a quote of said machine 840.

[0128] The VIM 870 causes the machine 840 to reboot.

[0129] The machine 840 reports to the VIM that it has successfully rebooted.

[0130] The VIM 870 requests that the attestation server 810 take a second quote of said machine 840.

[0131] The VIM 870 requests of the attestation server 810 its decision regarding the machine's 840 state, expected state, prior states and various combinations thereof as required.

[0132] Other subsequent actions may take place, for example:

[0133] The attestation server 810 reports that the machine 840 has failed attestation to the security orchestrator 830.

[0134] The security orchestrator 830 informs the VIM 870 that the security orchestrator 830 is taking actions to sand-box the machine by some means (e.g., virtual network function (VNF) wrapping, software defined networking (SDN) rerouting, etc.

[0135] Fig. 9 shows a flow chart of a process of an example embodiment. The process comprises:

[0136] 910. communicating with Internet devices 150 comprising trusted platform modules 152;

[0137] 920. checking states of a plurality of Internet devices 150 through quotes obtained from trusted platform modules 152 of the Internet devices 150;

[0138] 930. checking patch availability for each of the plurality of Internet devices 150;

[0139] 940. making patching decisions independent of other entities such as data processing agents in the update system or in communicative connection with the update system;

[0140] 950. determining the need for software updates based on quotes (e.g., by the patch management system 110); and

[0141] 960. maintaining policies that drive the quoting process.

[0142] In an example embodiment, the Internet devices 150 with a trusted platform module 152 can generate a hash that summarizes the hardware and software configuration of that platform, as well as other interesting metadata, such as reboot count. These hashes may be compared with a known good value in order to verify that a specific software is running on a machine. This process may be referred to as attestation of a platform. By storing the results of attestation into a database, either centralized or distributed, system agents may determine what software is running on each machine.

[0143] In an example embodiment, in cloud environments, such as the Telco Cloud, there are machines running the virtual workload, as well as different hardware connected to the cloud, such as base stations. The embodiments described in the foregoing may be used to find out the state in which these devices are and versions of software or firmware they are running.

[0144] Reference has been made to the measurements performed by the TPM 152. It should be appreciated that these measurements may also comprise version numbers of software and firmware or BIOS components.

[0145] As used in this application, the term “circuitry” may refer to one or more or all of the following:

(a) hardware-only circuit implementations (such as implementations in only analog and/or digital circuitry) and;

(b) combinations of hardware circuits and software, such as (as applicable):

(i) a combination of analog and/or digital hardware circuit(s) with software/firmware; and

(ii) any portions of hardware processor(s) with software (including digital signal processor(s)), software, and memory(ies) that work together to

cause an apparatus, such as a mobile phone or server, to perform various functions); and

(c) hardware circuit(s) and or processor(s), such as a microprocessor(s) or a portion of a microprocessor(s), that requires software (e.g., firmware) for operation, but the software may not be present when it is not needed for operation.

[0146] This definition of circuitry applies to all uses of this term in this application, including in any claims. As a further example, as used in this application, the term circuitry also covers an implementation of merely a hardware circuit or processor (or multiple processors) or portion of a hardware circuit or processor and its (or their) accompanying software and/or firmware. The term circuitry also covers, for example and if applicable to the particular claim element, a baseband integrated circuit or processor integrated circuit for a mobile device or a similar integrated circuit in server, a cellular network device, or other computing or network device.

[0147] Without in any way limiting the scope, interpretation, or application of the claims appearing below, a technical effect of one or more of the example embodiments disclosed herein is that reliability of patching Internet devices can be improved. Another technical effect of one or more of the example embodiments disclosed herein is that suitability of patches can be commonly tailored by owners or organizations that control numerous Internet devices. Yet another technical effect of one or more of the example embodiments disclosed herein is that actual performing of patching can be reliably verified by owners or controlling organizations. Yet another technical effect of one or more of the example embodiments disclosed herein is that patching can be improved for Internet of Things devices and other devices with insufficient direct manual supervision.

[0148] Embodiments of the present invention may be implemented in software, hardware, application logic or a combination of software, hardware and application logic. The software, application logic and/or hardware may reside on a patch management system, patch database, quote database, policy database and Internet device. If desired, part of the software, application logic and/or hardware may reside on a patch management system, and part of the software, application logic and/or hardware may reside on patch database, quote database, policy database and Internet device. In an example embodiment, the application logic, software or an instruction set is maintained on any one of various conventional computer-readable

media. In the context of this document, a "computer-readable medium" may be any non-transitory media or means that can contain, store, communicate, propagate or transport the instructions for use by or in connection with an instruction execution system, apparatus, or device, such as a computer, with one example of a computer described and depicted in Fig. 9. A computer-readable medium may comprise a computer-readable storage medium that may be any media or means that can contain or store the instructions for use by or in connection with an instruction execution system, apparatus, or device, such as a computer.

[0149] If desired, the different functions discussed herein may be performed in a different order and/or concurrently with each other. Furthermore, if desired, one or more of the before-described functions may be optional or may be combined.

[0150] Although various aspects of the invention are set out in the independent claims, other aspects of the invention comprise other combinations of features from the described embodiments and/or the dependent claims with the features of the independent claims, and not solely the combinations explicitly set out in the claims.

[0151] It is also noted herein that while the foregoing describes example embodiments of the invention, these descriptions should not be viewed in a limiting sense. Rather, there are several variations and modifications which may be made without departing from the scope of the present invention as defined in the appended claims.

CLAIMS

1. An update system, comprising:

a communication block configured to communicate with the Internet devices;

computer program code; and

a processing block configured to execute the computer program code and accordingly to perform:

checking states of a plurality of Internet devices through quotes obtained from trusted platform modules of the Internet devices;

checking patch availability for each of the plurality of Internet devices;

making patching decisions independent of other entities;

determining the need for software updates based on quotes; and

maintaining policies that drive the quoting process.

2. The update system of claim 1, wherein the checking of the states of the plurality of Internet device comprises checking for each of the Internet devices an update state for each of one or more updateable components.

3. The update system of claim 1 or 2, wherein the checking of the update state may comprise checking of current software or firmware version.

4. The update system of any one of preceding claims, wherein the communication block comprises computer program code.

5. The update system of any one of preceding claims, wherein the communication block comprises a communication circuitry.

6. The update system of any one of preceding claims, wherein the processing block comprises computer program code.

7. The update system of any one of preceding claims, wherein the processing block comprises a processing circuitry.

8. The update system of any one of preceding claims, wherein the received quotes comprise platform configuration registers.

9. The update system of any one of preceding claims, wherein the processing block is further configured to define the policies that drive the quoting process.

10. The update system of any one of preceding claims, wherein the update system comprises a patch management system.

11. The update system of claim 10, wherein the patch management system comprises the communication block; the processing block; and at least some of the

computer program code.

12. The update system of claim 11, wherein the determining of the need for software updates based on the quotes is made by the patch management system.

13. The update system of any one of preceding claims, wherein the update system further comprises a policy database comprising the policies that drive the quoting process.

14. The update system of any one of preceding claims, wherein the update system further comprises a quote engine configured to provide the patch management system with the predicted quotes.

15. The update system of claim 14, wherein the quote engine comprises a quote database comprising previously computed predicted quotes for different Internet device components and patches.

16. The update system of claim 14 or 15, wherein the quote engine further comprises a quote processing block configured to compute predicted quotes on request of the patch management system.

17. The update system of any one of preceding claims, wherein the updating system further comprises a patch database.

18. The update system of claim 17, wherein the patch database comprises patches for the components of the Internet devices.

19. The update system of claim 17 or 18, wherein the patch database comprises links to externally stored patches for the components of the Internet devices.

20. The update system of any one of preceding claims, wherein the Internet device comprises updatable software elements.

21. The update system of claim 20, wherein the updateable software elements comprise a basic input/output system program code.

22. The update system of claim 20 or 21, wherein the updateable software elements comprise firmware of a hardware component of the Internet device.

23. The update system of any one of claims 20 to 22, wherein the software element comprises a device driver.

24. The update system of any one of claims 20 to 23, wherein the software element comprises at least one operating system component.

25. The update system of any one of claims 20 to 24, wherein the software element comprises at least one application component.

26. The update system of any one of preceding claims, wherein the processing block

is further configured to perform verifying each of the Internet devices has been successfully patched.

27. The update system of claim 26, wherein the verifying comprises comparing a received quote of the Internet device with a predicted quote.

28. The update system of claim 26 or 27, wherein the predicted quote comprises check a code produced for a software element being patched after a patch in question has been applied to the software element.

29. The update system of any one of claims 26 to 28, wherein the check code comprises an authenticator.

30. The update system of claim 29, wherein the authenticator comprises a cryptographic presentation of the check code.

31. The update system of claim 30, wherein the cryptographic presentation comprises a cryptographic one-way hashing of the check code.

32. The update system of any one of claims 29 to 31, wherein the authenticator comprises a digital signature of the check code.

33. The update system of claim 32, wherein the digital signature is formed using an attestation key of the Internet device.

34. The update system of claim 33, wherein the attestation key is formed by a trusted platform module of the Internet device.

35. The update system of any one of claims 29 to 34, wherein the authenticator comprises as such or as a derivative state information of the trusted platform module.

36. The update system of any one of claims 29 to 35, wherein the authenticator comprises as such or as a derivative a timestamp.

37. The update system of any one of claims 26 to 36, wherein the verifying of successful patching comprises obtaining a new quote from an Internet device and determining whether the new quote matches a predicted quote.

38. The update system of any one of preceding claims, wherein the update system comprises an attestation server.

39. The update system of claim 38, wherein the patch management system is configured to communicate directly with the Internet devices.

40. The update system of claim 38, wherein the patch management system is configured to communicate directly with the Internet devices via the attestation server.

41. The update system of any one of preceding claims, wherein the update system comprises a blockchain delivery channel configured to distribute patches to the

Internet devices through a blockchain.

42. The update system of any one of preceding claims, wherein any one of the policy database, quote database; and patch database is centralized or distributed.

43. The update system of claim 42, wherein the centralized databases are implemented using a relational database, such as a structured query language database.

44. The update system of claim 42 or 43, wherein the distributed databases are implemented using one or more distributed relational databases and/or a blockchain.

45. The update system of any one of preceding claims, wherein the update system comprises a subset of the plurality of Internet devices.

46. The update system of any one of preceding claims, wherein the entities comprise data processing agents in communicative connection with the update system.

47. The update system of any one of preceding claims, wherein the entities comprise data processing agents in the update system.

48. A method comprising:

communicating with Internet devices comprising trusted platform modules;

checking states of a plurality of Internet devices through quotes obtained from trusted platform modules of the Internet devices;

checking patch availability for each of the plurality of Internet devices;

making patching decisions independent of other entities;

allowing the patch management system to determine the need for software updates based on quotes; and

maintaining policies that drive the quoting process.

49. A computer program comprising computer executable program code configured to execute the method of claim 48.

50. A non-transitory memory medium comprising the computer program of claim 49.

51. An apparatus, comprising:

a communication block configured to communicate with the Internet devices;

computer program code; and

a processing block configured to execute the computer program code and accordingly to perform:

checking states of a plurality of Internet devices through quotes obtained from trusted platform modules of the Internet devices;

checking patch availability for each of the plurality of Internet devices;
making patching decisions independent of other entities;
determining the need for software updates based on quotes; and
maintaining policies that drive the quoting process.

5

Fig. 1

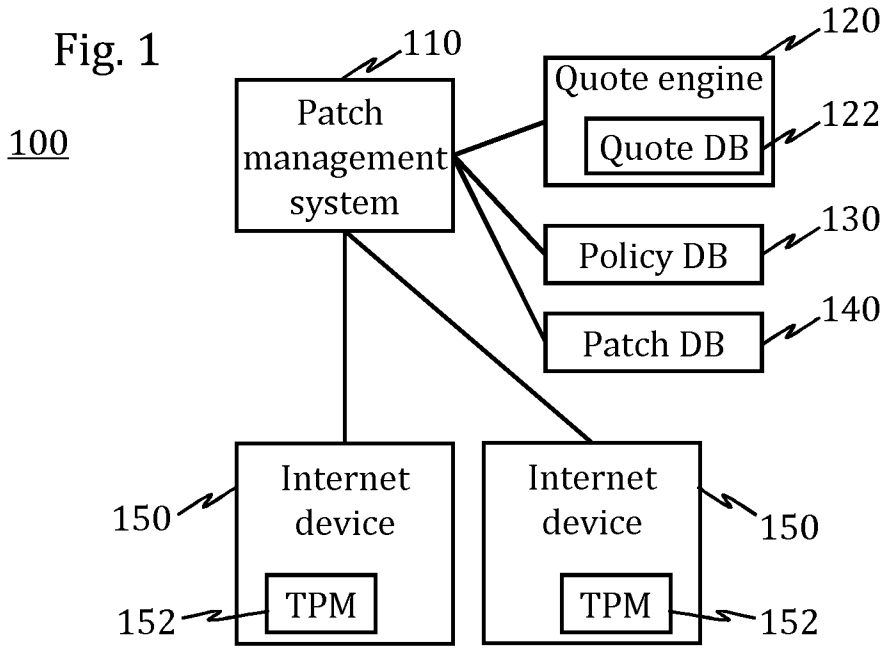


Fig. 2

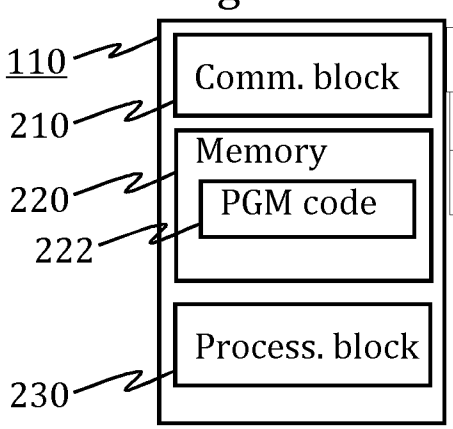


Fig. 3

Fig. 4

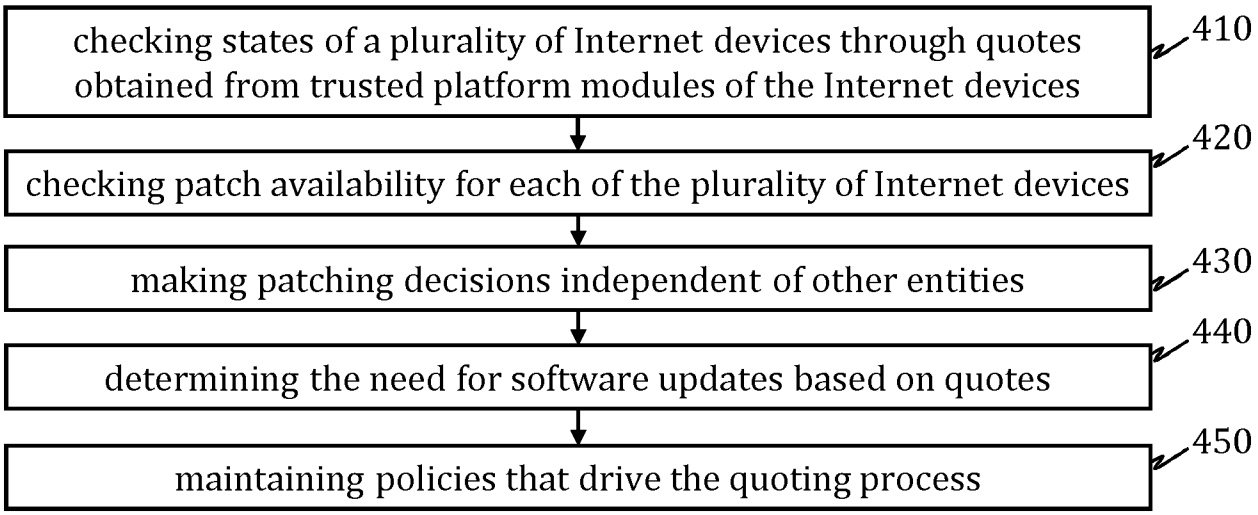


Fig. 5

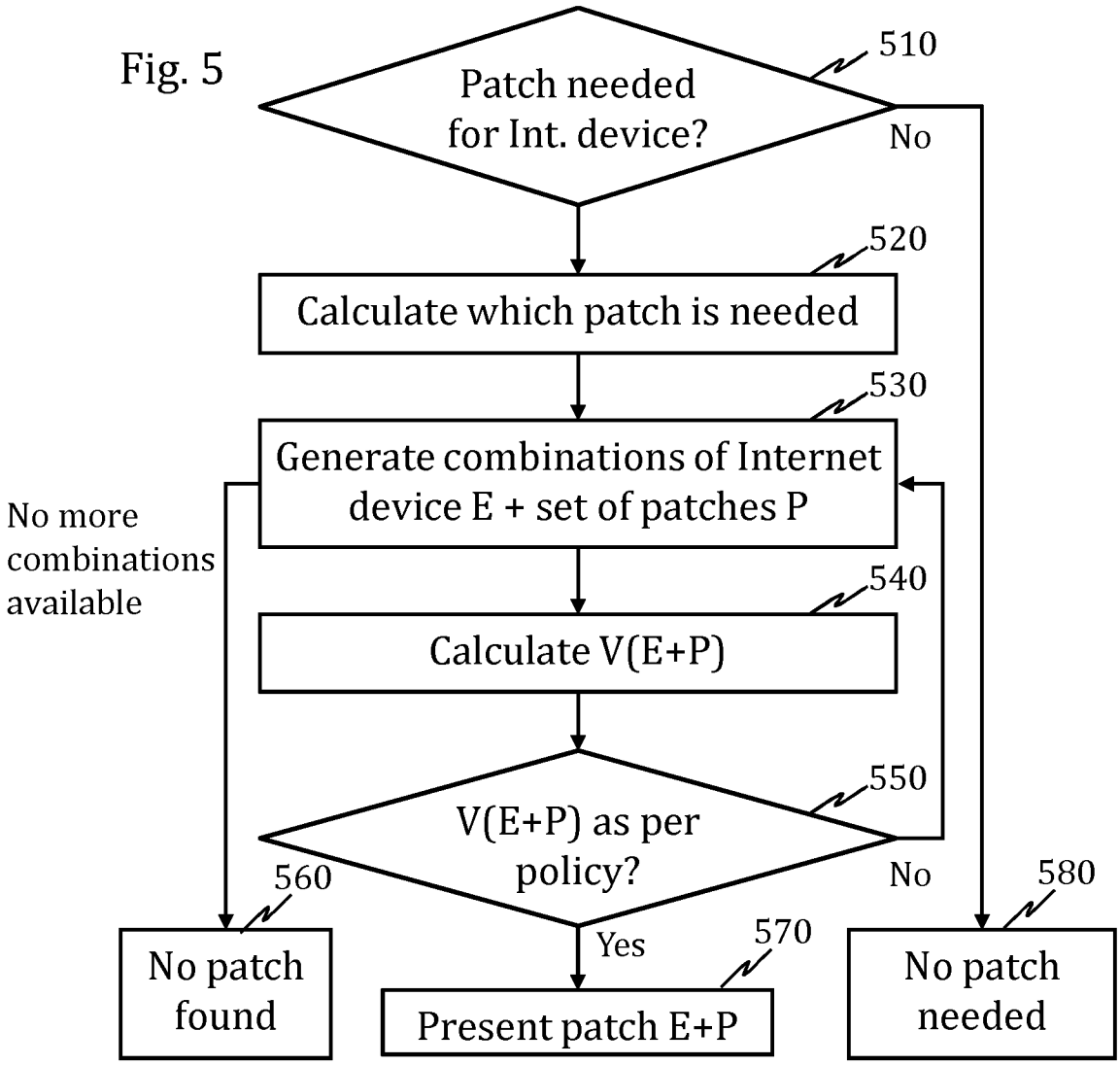
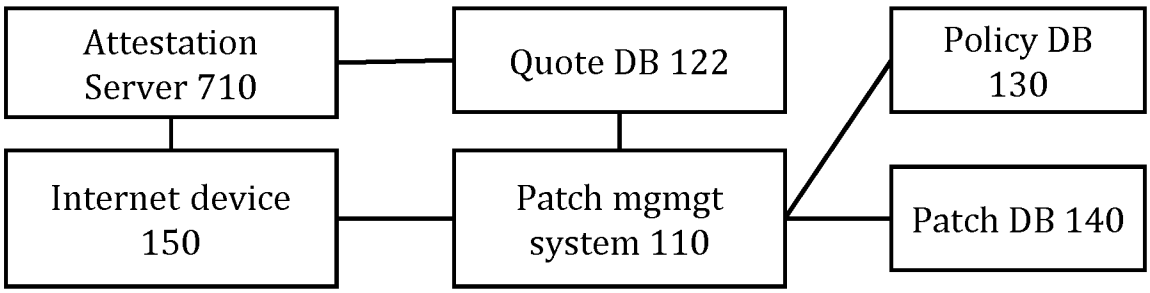

Patch needed for Int. device?
510
No
520
Calculate which patch is needed
530
Generate combinations of Internet device E + set of patches P
No more combinations available
540
Calculate V(E+P)
550
V(E+P) as per policy?
No
Yes
560
No patch found
570
Present patch E+P
580
No patch needed

Fig. 7


Attestation Server 710
Quote DB 122
Policy DB 130
Internet device 150
Patch mgmgt system 110
Patch DB 140

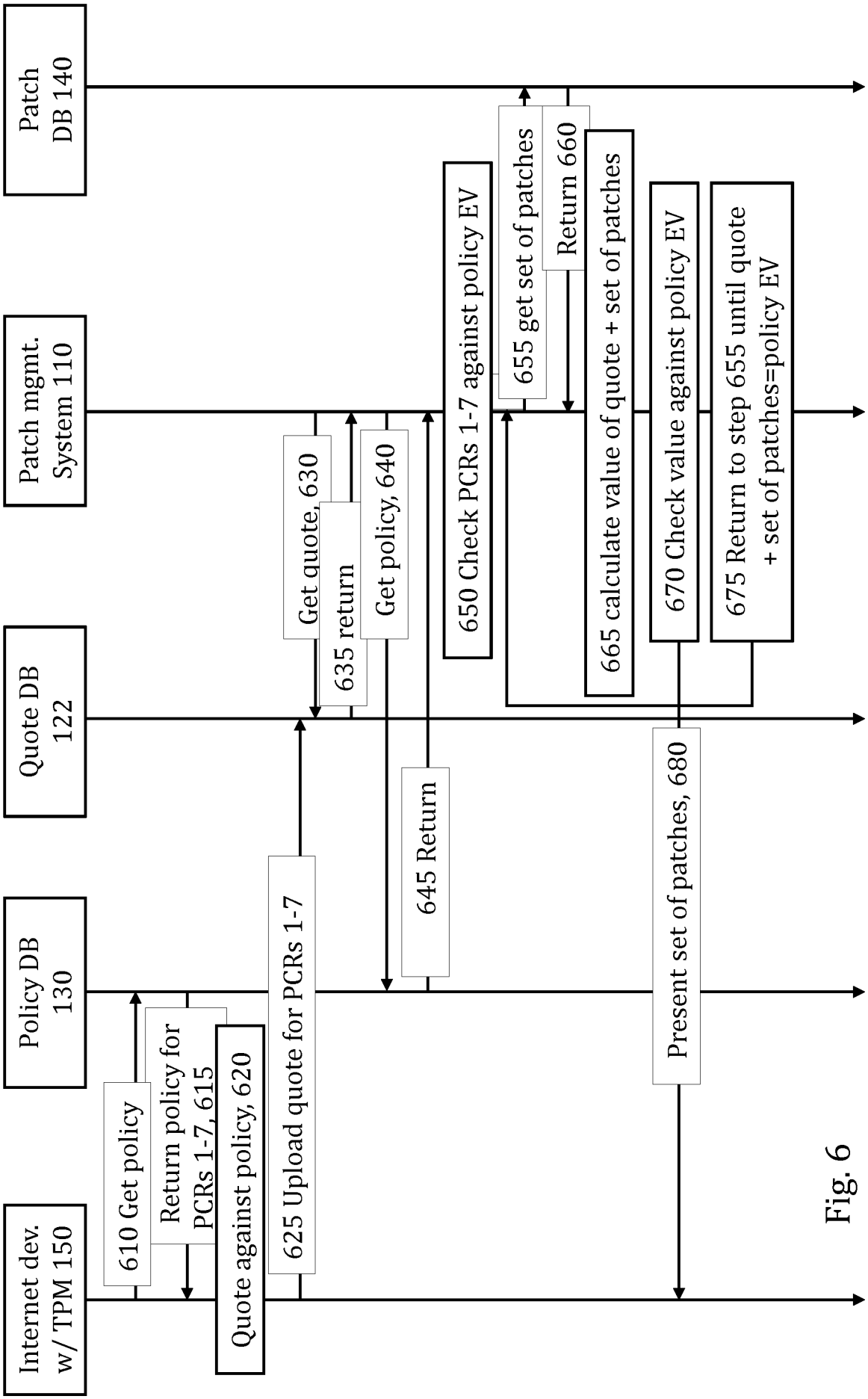

Internet dev. w/ TPM 150
Policy DB 130
Quote DB 122
Patch mgmt. System 110
Patch DB 140
610 Get policy
Return policy for PCRs 1-7, 615
Quote against policy, 620
625 Upload quote for PCRs 1-7
Get quote, 630
635 return
Get policy, 640
645 Return
650 Check PCRs 1-7 against policy EV
655 get set of patches
Return 660
665 calculate value of quote + set of patches
670 Check value against policy EV
675 Return to step 655 until quote + set of patches=policy EV
Present set of patches, 680

Fig. 6

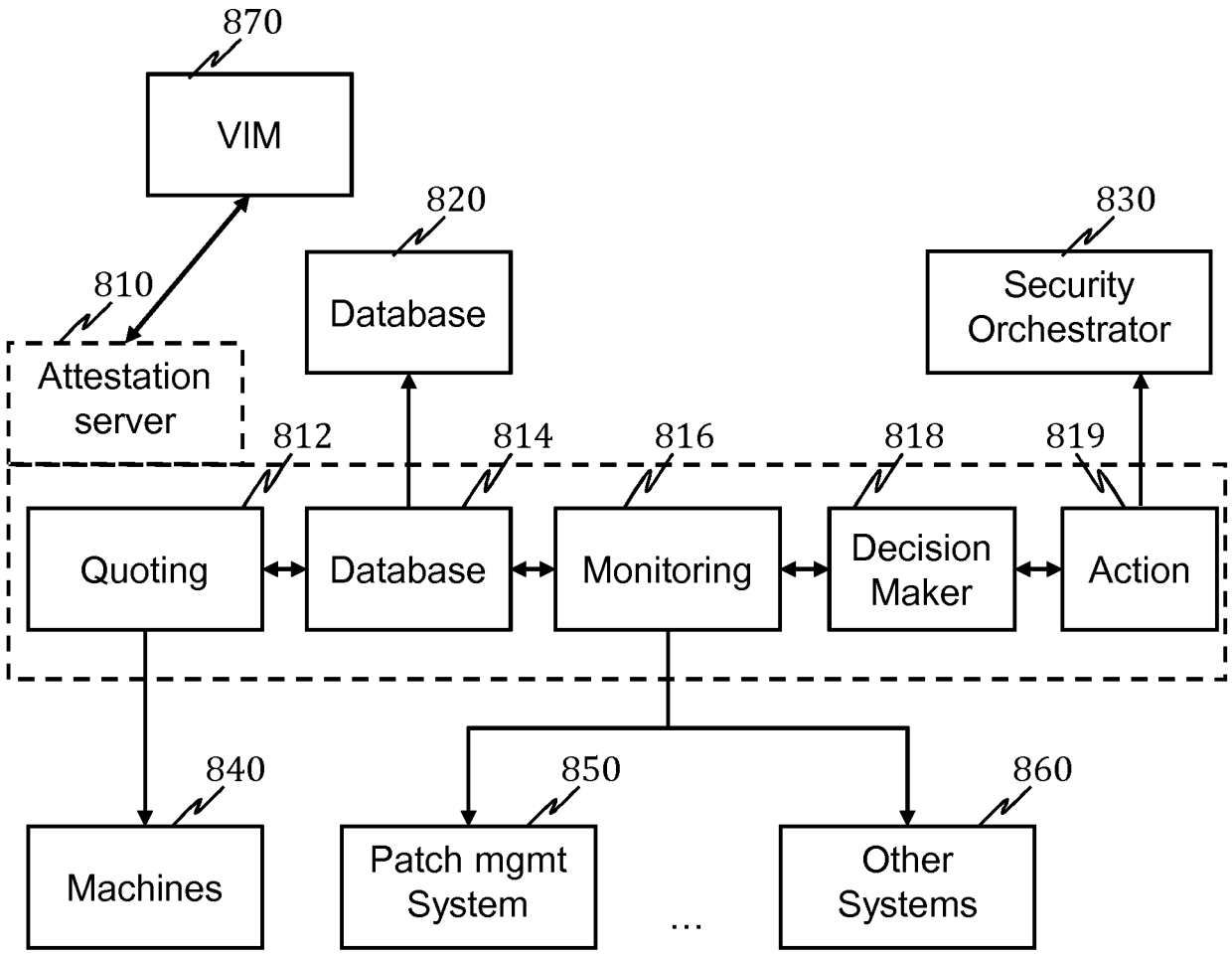
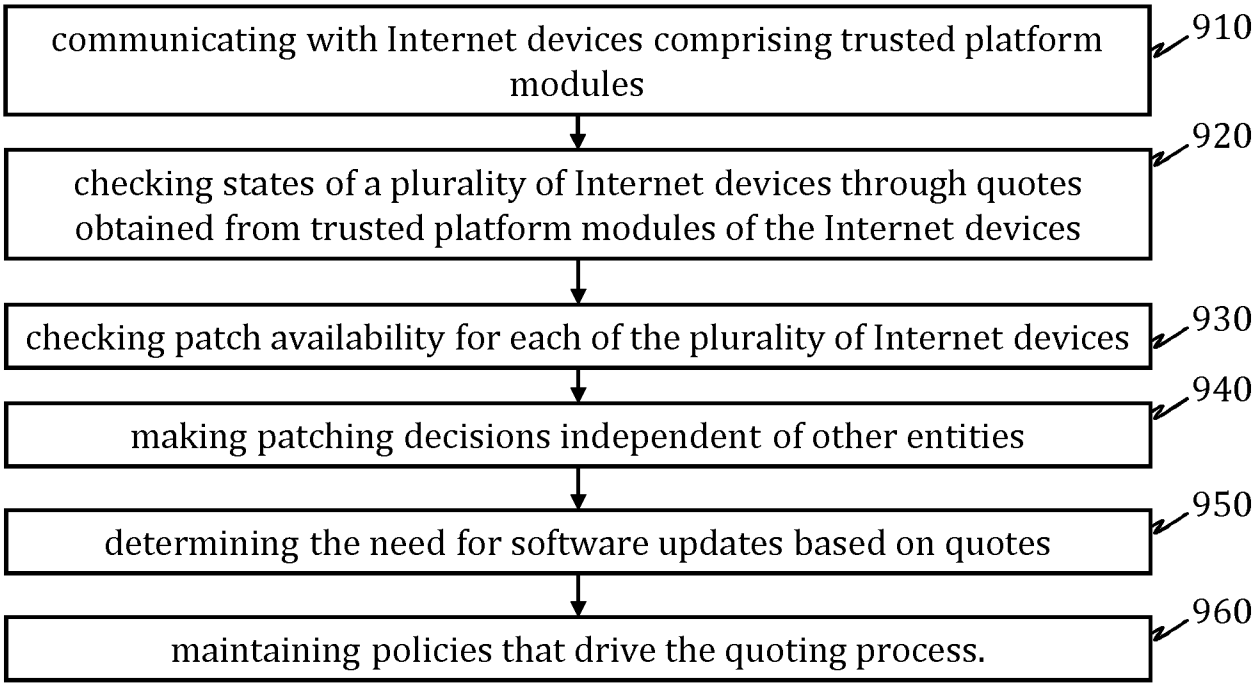

870
VIM
820
Database
830
Security Orchestrator
810
Attestation server
812
814
816
818
819
Quoting
Database
Monitoring
Decision Maker
Action
840
850
860
Machines
Patch mgmt System
...
Other Systems

Fig. 8

Fig. 9


communicating with Internet devices comprising trusted platform modules
910
checking states of a plurality of Internet devices through quotes obtained from trusted platform modules of the Internet devices
920
checking patch availability for each of the plurality of Internet devices
930
making patching decisions independent of other entities
940
determining the need for software updates based on quotes
950
maintaining policies that drive the quoting process.
960

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI2018/050647

A. **CLASSIFICATION OF SUBJECT MATTER**

See extra sheet

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

FI, SE, NO, DK

Electronic data base consulted during the international search (name of data base, and, where practicable, search terms used)

EPODOC, EPO-Internal full-text databases, Full-text translation databases from Asian languages, WPIAP, XPAIP, XPESP, XPI3E, XPIEE, XPIOP, XPIPCOM, XPMISC, XPOAC, XPRD, XPTK, COMPDX, INSPEC, TDB, NPL

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2018007040 A1 (THOM STEFAN [US] et al.) 04 January 2018 (04.01.2018) Abstract; Figs. 2, 5; paragraphs [0001], [0005], [0016], [0017], [0019], [0020], [0022], [0026], [0033]-[0049], [0084], [0085]	1-51
A	US 2018246709 A1 (GILBERT DAVID A [GB] et al.) 30 August 2018 (30.08.2018) Paragraphs [0005]-[0009]	
A	US 2018114000 A1 (TAYLOR RICHARD MICHAEL [GB]) 26 April 2018 (26.04.2018) Abstract; claim 1	

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
10 January 2019 (10.01.2019)	11 January 2019 (11.01.2019)

Name and mailing address of the ISA/FI
Finnish Patent and Registration Office
FI-00091 PRH, FINLAND
Facsimile No. +358 29 509 5328

Authorized officer
Vesa-Matti Louekoski
Telephone No. +358 29 509 5000

INTERNATIONAL SEARCH REPORT
Information on Patent Family Members

International application No.
PCT/FI2018/050647

Patent document cited in search report	Publication date	Patent family members(s)	Publication date
US 2018007040 A1	04/01/2018	WO 2018005248 A1	04/01/2018
US 2018246709 A1	30/08/2018	CN 103329093 A	25/09/2013
		CN 103329093 B	12/09/2017
		DE 112012000512 T5	24/10/2013
		GB 201313795 D0	18/09/2013
		GB 2501433 A	23/10/2013
		GB 2501433 B	04/06/2014
		JP 2014503101 A	06/02/2014
		JP 5932837 B2	08/06/2016
		KR 20130114672 A	17/10/2013
		US 2014026124 A1	23/01/2014
		US 9317276 B2	19/04/2016
		US 2016162285 A1	09/06/2016
		US 10007510 B2	26/06/2018
		US 2016162396 A1	09/06/2016
		US 10108413 B2	23/10/2018
		WO 2012098478 A1	26/07/2012
US 2018114000 A1	26/04/2018	EP 3295352 A1	21/03/2018
		GB 201508035 D0	24/06/2015
		WO 2016181152 A1	17/11/2016

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI2018/050647

CLASSIFICATION OF SUBJECT MATTER

IPC
G06F 8/65 (2018.01)
G06F 21/57 (2013.01)